UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/891,178	08/03/2001	Peter C. Jones	06502.0062-02	2575

22852 7590 04/21/2005

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
COULTER, KENNETH R	

ART UNIT	PAPER NUMBER
2141	

DATE MAILED: 04/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/891,178	JONES ET AL.	
	Examiner	Art Unit	
	Kenneth R. Coulter	2141	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 October 2004 (Appeal Brief)*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *21-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *21-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 December 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. In view of the Supplemental Appeal Brief filed on 10/14/2004, PROSECUTION IS HEREBY REOPENED. New grounds of rejection are set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) request reinstatement of the appeal.

If reinstatement of the appeal is requested, such request must be accompanied by a supplemental appeal brief, but no new amendments, affidavits (37 CFR 1.130, 1.131 or 1.132) or other evidence are permitted. See 37 CFR 1.193(b)(2).

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 21 - 24, 32, 33, 39, and 40 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

3. Regarding claim 21, 23, 32, 39, and 40, applicant discloses in the preamble "used by a **program**".

These claims represent a computer program that is not explicitly implemented on a storage medium, and therefore is directed to non-statutory subject matter.

The following passages from **MPEP 2106** are relevant.

(a) Functional Descriptive Material: "Data Structures" Representing Descriptive Material Per Se or Computer Programs Representing Computer Listings Per Se

Data structures not claimed as embodied in computer-readable media are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer. See, e.g., Warmerdam, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory.

Similarly, computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such **claimed computer programs do not define any structural and functional interrelationships** between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. **In contrast, a claimed computer- readable medium encoded with a computer program** is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus **statutory**.

Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 27 – 29 and 36 – 38 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

The feature of "receiving a stream containing an identifier of an event listener" is purportedly shown in Fig. 9, 901 (see Appeal Brief filed on 10/14/2004; p. 5, lines 12 – 14).

The Examiner does not see this feature in Figure 9, item 901.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 21 – 42 are rejected under 35 U.S.C. 102(e) as being anticipated by Moore et al. (U.S. Pat. No. 6,408;342) (Communications Framework for Supporting Multiple Simultaneous Communications Protocols in a Distributed Object Environment).

6.1 Regarding claim 21, Moore discloses a data processing system having an RPC mechanism used by a program, a method for transmitting objects comprising:

receiving an object in a form of a stream from a remote RPC mechanism (Figs. 1, 5, 6, and 8; col. 4, lines 39 – 51 (see below)); and

The marshaling and demarshaling of arguments passed to remote methods is accomplished according to the invention by defining an Out**Stream** class. The Out**Stream** class defines an interface for at least one primitive marshaler and for a composite data type marshaler, wherein each remote procedure call transport derives an OutStream object from the OutStream class for marshaling arguments onto the communications link. The communication framework also includes a composite data type class and at least one transport independent marshaler. The OutStream object recognizes any argument that is of a composite data type. The RPC_Transport invokes a transport independent marshaler to marshal any composite data type argument objects.

deferring reconstruction of the object until requested to perform reconstruction by the program (col. 22, lines 32 – 44 (see below)).

The communications framework creates a new ObjectReference 501 for a target object whenever a target object is first registered with the communications framework. Optionally, **the construction of the ObjectReference 501 may be *delayed until it is needed***, thus avoiding any unnecessary ObjectReference 501 creation. The created **ObjectReference 501** is **passed to other processes** either by returning the ObjectReference 501 as a **return parameter from a remote procedure call** to another process, or by **passing the ObjectReference 501 as a parameter** in an **outbound remote procedure call**. Alternatively, the ObjectReference 501 can be made known to other processes by placing it in a shared medium, such as a shared disk file.

6.2 Per claim 22, Moore teaches the method of claim 21, further comprising:

reconstructing the object using code identified in the stream, when requested to perform reconstruction by the program (col. 22, lines 32 – 44; col. 16, lines 45 – 50 (see below)).

Demarshaling is an analogous process to marshaling. Each object associated with a transmittable value supports a demarshal() routine. The **demarshal() routine derives from the InStream** class 409.

6.3 Regarding claim 23, Moore discloses a method in a data processing system for transmitting an object from a first RPC mechanism to a second RPC mechanism that is used by a program, comprising:

forming a stream out of the object by the first RPC mechanism (Figs. 1, 5, 6, and 8; col. 4, lines 39 – 51);

sending the stream to the second RPC mechanism by the first RPC mechanism (Fig. 5, item 101);

receiving the stream by the second RPC mechanism (Fig. 5, item 103); and

deferring reconstruction of the object by the second RPC mechanism until requested to perform the reconstruction by the program (col. 22, lines 32 – 44).

The communications framework creates a new ObjectReference 501 for a target object whenever a target object is first registered with the communications framework. Optionally, **the construction of the ObjectReference 501 may be *delayed until it is needed***, thus avoiding any unnecessary ObjectReference 501 creation. The created **ObjectReference 501** is **passed to other processes** either by returning the ObjectReference 501 as a **return parameter from a remote procedure call** to another process, or by **passing the ObjectReference 501 as a parameter** in an **outbound remote procedure call**. Alternatively, the ObjectReference 501 can be made known to other processes by placing it in a shared medium, such as a shared disk file.

6.4 Per claim 24, Moore teaches the method of claim 23, further comprising the step, performed by the second RPC mechanism, of:

reconstructing the object using code identified in the stream, when requested to perform reconstruction by the program (Figs. 1, 5, 6, and 8; col. 4, lines 39 – 51).

6.5 Regarding claim 25, Moore discloses a method in a data processing system for transmitting an object from a first RPC mechanism to a second RPC mechanism, comprising:

forming a stream out of the object by the first RPC mechanism (Figs. 1, 5, 6, and 8; col. 4, lines 39 – 51);

sending the stream from the first RPC mechanism to the second RPC mechanism (Fig. 5, item 101);

storing the stream by the second RPC mechanism (col. 16, lines 34 – 41 (see below)); and

The marshaling and demarshaling mechanism of the present invention places no restriction on the **in-memory** representation of an object. The only requirement is that each marshalable object supports a marshal() and demarshal() method (either directly or in the preferred embodiment via a base class). In the above example, the object provides its own marshaler.

deferring reconstruction of the object by the first RPC mechanism until the stream is **returned** from the second RPC mechanism to the first RPC mechanism in response to the occurrence of an event (col. 11 lines 32 – 44 (see below)).

The communications framework creates a new ObjectReference 501 for a target object whenever a target object is first registered with the communications framework. Optionally, **the construction of the ObjectReference 501 may be *delayed until it is needed***, thus avoiding any unnecessary ObjectReference 501 creation. The created **ObjectReference 501** is **passed to other processes** either by returning the ObjectReference 501 as a ***return parameter*** **from a remote procedure call** to another

process, or by **passing the ObjectReference 501 as a parameter** in an **outbound remote procedure call**. Alternatively, the ObjectReference 501 can be made known to other processes by placing it in a shared medium, such as a shared disk file.

6.6 Per claim 26, Moore teaches the method of claim 25, further comprising:

reconstructing the object by the first RPC mechanism using code identified in the stream (col. 22, lines 32 – 44; col. 16, lines 45 – 50 (see below)).

Demarshaling is an analogous process to marshaling. Each object associated with a transmittable value supports a demarshal() routine. The **demarshal() routine derives from the InStream** class 409.

6.7 Regarding claim 27, Moore discloses a method for processing objects in a distributed system comprised of multiple machines, comprising:

receiving a stream containing an identifier of an event listener and a self-describing form of an object associated with a request for notification of a particular event within the distributed system (Figs. 4B, 5; col. 25, lines 52 – 60 (see below)); and

The RPC_Transport 305 for each communication protocol includes a **listener** to receive incoming requests for the physical media supported by the protocol. In the example of FIG. 5, the listener is the RPC_Server 315. When the **listener demarshals** (calling upon the primitive marshalers 313) the object identifier, the Virtual Process identifier, and the operation name associated with the incoming request. The RPC_Transport 305 uses these pieces of information to create an IncomingCall instance derived from the following IncomingCall class:

in response to occurrence of the particular *event*, sending the stream to the identified event listener for reconstruction of the object using program code identified in the stream (col. 25, lines 52 - 60).

6.8 Per claim 28, Moore teaches the method of claim 27, wherein the stream is received from the event listener (Figs. 4B, 5; col. 25, lines 52 – 60).

6.9 Regarding claim 29, Moore discloses the method of claim 27, wherein the stream is received from a machine **other than** the event listener (Figs. 4B, 5, 6, 7; col. 25, lines 52 – 60).

6.10 Per claims 30 – 35 and 39 – 42, the rejection of claims 21 – 26 under 35 USC 102(e) (paragraphs 6.1 – 6.6 above) applies fully.

6.11 Regarding claims 36, 37, and 38, the rejection of claims 27, 28, and 29 respectively under 35 USC 102(e) (paragraphs 6.7 – 6.9 above) applies fully.

7. Claims 21 – 26, 30 – 35, and 39 – 42 are rejected under 35 U.S.C. 102(e) as being disclosed by Heimsoth et al. (Object-Oriented Communication Interface for Network Protocol Access Using the Selected Newly Created Protocol Interface Object and Newly Created Protocol Layer Objects in the Protocol Stack).

7.1 Regarding claim 21, Heimsoth discloses a data processing system having an RPC mechanism used by a program, a method for transmitting objects comprising:

receiving an object in a form of a stream from a remote RPC mechanism (Fig. 9D; col. 30, lines 1 – 10 "This function also **rebuilds** the NetworkOperation ***object*** **when the server responds to the request that was sent**."; col. 29, lines 41 – 46; col. 31, lines 5 - 18); and

deferring reconstruction of the object until requested to perform reconstruction by the program (Fig. 9D; col. 29, lines 41 – 46; col. 31, lines 5 - 18).

7.2 Per claim 22, Heimsoth teaches reconstructing the object using code identified in the stream, when requested to perform reconstruction by the program (Fig. 9D; col. 30, lines 1 – 10; col. 29, lines 41 – 46; col. 31, lines 5 - 18).

7.3 Regarding claim 23, Heimsoth discloses a method in a data processing system for transmitting an object from a first RPC mechanism to a second RPC mechanism that is used by a program, comprising:

forming a stream out of the object by the first RPC mechanism (col. 29, lines 26 - 30);

sending the stream to the second RPC mechanism by the first RPC mechanism (Fig. 7A; col. 30, lines 1 – 10; col. 29, lines 41 – 46; col. 31, lines 5 - 18);

receiving the stream by the second RPC mechanism (Fig. 7A; col. 30, lines 1 – 10; col. 29, lines 41 – 46; col. 31, lines 5 - 18); and

deferring reconstruction of the object by the second RPC mechanism until requested to perform the reconstruction by the program (Fig. 9D; col. 30, lines 1 – 10; col. 29, lines 41 – 46; col. 31, lines 5 - 18).

7.4 Per claim 24, Heimsoth teaches the method of claim 23, further comprising the step, performed by the second RPC mechanism, of:

reconstructing the object using code identified in the stream, when requested to perform reconstruction by the program (Fig. 9D; col. 30, lines 1 – 10; col. 29, lines 41 – 46; col. 31, lines 5 - 18).

7.5 Regarding claim 25, Heimsoth discloses a method in a data processing system for transmitting an object from a first RPC mechanism to a second RPC mechanism, comprising:

forming a stream out of the object by the first RPC mechanism (Fig. 9D; col. 29, lines 26 – 30 and 41 – 46; col. 31, lines 5 - 18);

sending the stream from the first RPC mechanism to the second RPC mechanism (Fig. 7A; col. 29, lines 41 – 46; col. 31, lines 5 - 18);

deferring reconstruction of the object by the first RPC mechanism until the stream is returned from the second RPC mechanism to the first RPC mechanism in

response to the occurrence of an event (Fig. 9D; col. 30, lines 1 – 10; col. 29, lines 41 – 46; col. 31, lines 5 - 18).

7.6 Per claim 26, Heimsoth teaches the method of claim 25, further comprising: reconstructing the object by the first RPC mechanism using code identified in the stream (col. 29, lines 26 - 30).

7.7 Per claims 30 – 35 and 39 – 42, the rejection of claims 21 - 26 under 35 USC 102(e) (paragraphs 7.1 – 7.6 above) applies fully.

Response to Arguments

8. Applicant's arguments filed 10/14/2004 (Supplemental Appeal Brief) have been fully considered but they are not persuasive.

The Examiner notes that Applicant is incorrect in stating that "the Examiner did not properly address all of the recitations of claims 23 - 26, 32 - 35, and 39 - 42 when rejecting these claims but simply relied on the reasons set forth in the rejection of claims 21 and 22." (p. 7 of the Appeal Brief filed 10/14/2004).

Claim 25 was explicitly rejected on 1/30/2004 (p. 3, item 2.3).

Applicant states that "Heimsoth et al. does not teach or suggest deferring the reconstruction of an object." (p. 7; Supplemental Appeal Brief filed on 10/14/2004).

Examiner disagrees.

Heimsoth discloses a ProcessOperation function that "rebuilds the NetworkOperation object when the server responds to the request that was sent." (col. 30, lines 1 – 10). Claim 21 is indistinguishable from conventional systems since the claimed deferring process does not express a reason for deferring or a timetable for deferring. Claim 21 merely states that reconstruction of the object is deferred until a program requests the reconstruction.

Applicant states that "Examiner did not properly address all of the recitations of claims 23 – 26, 32 – 35, and 39 – 42." (p. 7; Supplemental Appeal Brief filed on 10/14/2004). A more detailed rejection with regard to Heimsoth is given above.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth R. Coulter whose telephone number is 571 272-3879. The examiner can normally be reached on 5 4 9.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rupal Dharia can be reached on 571 272-3880. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KENNETH R. COULTER
PRIMARY EXAMINER

krc